UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
NPRDC TR 87-8

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Navy Personnel Research and Development Center

6b. OFFICE SYMBOL *(If applicable)*
Code 41

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
San Diego, CA 92152-6800

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Office of Naval Technology

8b. OFFICE SYMBOL *(If applicable)*
Code 222

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*
Washington, DC 22217-5000

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO.	WORK UNIT ACCESSION NO.
62766N	NP2A	1	

11. TITLE *(Include Security Classification)*
REAL-TIME FAULT DETECTION AND DIASNOSIS: THE USE OF LEARNING EXPERT SYSTEMS TO HANDLE THE TIMING OF EVENTS

12. PERSONAL AUTHOR(S)
Donald B. Malkoff

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Technical Report	FROM 84 Oct TO 86 Sep	1986 November	41

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Artificial intelligence, real-time systems, expert systems, multisensor integration, fault detection and diagnosis, temporal data,

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The successful performance of real-time, sensor-based fault detection and diagnosis in large and complex systems is seldom achieved by operators. Examples of operator and system failures are presented and analyzed. The lack of an effective method for handling temporal data is seen as one of the key problems in this area. As part of the solution to these problems, a methodology is introduced that is able to make good use of temporal data to perform fault diagnosis in a subsystem of a Navy ship gas turbine engine propulsion unit. The methodology is embedded in a computer program designed to be used as a decision aid to assist the operator. It utilizes machine learning, is able to cope with uncertainty at several levels, works in real-time, and is developed to the point of possible application. Data are presented and analyzed with regard to the effectiveness of this approach. Relevance and applicability to other process control and classification problems are discussed. The approach is put forth as an example of how relatively simple existing techniques can be assembled into more powerful real-time diagnostic tools.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Donald B. Malkoff

22b. TELEPHONE *(Include Area Code)*
(619) 225-6617

22c. OFFICE SYMBOL
Code 41

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Block 18 (Continued)

stochastic processes, adaptable programming, gas turbine engines, man-machine interface, learning machines.

FOREWORD

The research presented in this report was conducted with financial support provided from the Navy Exploratory Development Plan; Personnel, Training, and Human Factors block for the Human Factors Project (NP2A), Task 1, Expert Systems Technology for Real-time Propulsion Control. Additional support was provided under Independent Exploratory Development program element 62766N, Expert Systems for Fault Diagnosis: Stochastic Processes.

The objective of the work is to explore and develop computer technologies to reduce the task complexity operators experience in coping with malfunctions during the operation of complex and critical real-time control systems. The current effort describes a machine learning methodology developed and implemented in a laboratory setting using, as a test-bed, a simulation of the reduction gear lubrication oil subsystem of the surface ship gas turbine propulsion plant. The methodology is embedded in computer programs collectively designed to be used as a decision aid to assist the propulsion plant operator. It utilizes machine learning, is able to cope with uncertainty at several levels, and is developed beyond the point of a research tool. Data are presented and analyzed with regard to the effectiveness of the approach. Relevance and applicability to other process control and classification problems are discussed.

B. E. BACON
Captain, U.S. Navy
Commanding Officer

J. W. TWEEDDALE
Technical Director

NPRDC-TR 87-8

NOVEMBER 1986

REAL-TIME FAULT DETECTION AND DIAGNOSIS: THE USE OF LEARNING EXPERT SYSTEMS TO HANDLE THE TIMING OF EVENTS

NAVY PERSONNEL RESEARCH
AND
DEVELOPMENT CENTER
San Diego, California 92152

NPRDC TR 87-8

November 1986

REAL-TIME FAULT DETECTION AND DIAGNOSIS: THE USE OF LEARNING EXPERT SYSTEMS TO HANDLE THE TIMING OF EVENTS

Donald B. Malkoff

Reviewed by
E. A. Koehler

Approved by
R. E. Blanchard

Released by
B. E. Bacon
Captain, U.S. Navy
Commanding Officer

Navy Personnel Research and Development Center
San Diego, California 92152-6800

SUMMARY

Problem

Personnel assigned the task of controlling large, complex real-time systems have considerable difficulties in correctly diagnosing system malfunctions. This is particularly the case in systems where the operator must make decisions based upon sets of alarms triggered by sensor-data whose values have exceeded specified operational range-limits. Information overload is almost always a significant part of this problem. However, in many of these systems there is insufficient information for diagnosis regardless of the volume of data or the real-time constraints of the system. Many distinctly different causes of malfunctions result in identical sets of alarms. In such cases, operators are completely unable to render meaningful decisions.

Objective

In these systems, there is often valuable additional information available that could be used to enhance the operators diagnostic capabilities. The sequential order in which alarms are triggered is an example of readily available, easily employed, highly beneficial information that is, unfortunately, seldom used for this purpose. Even more powerful is the information represented by the time at which these alarms are triggered. The latter is not easily used to its full advantage. The fact that (1) the timing of system events is subject to considerable natural variability and (2) the state of both the system and its environment are constantly changing presents a formidable obstacle to the development of a method for effectively using time for the purpose of decision-making. The objective of this work is to develop and evaluate a methodology for successfully doing this, utilizing learning machine principles to overcome the problems of continual changes in the system and in the sensor inputs. The method must work in real-time and in the face of incomplete information, providing predictions as to the cause of a malfunction as it is in the course of unfolding.

Approach

A portion of the Navy surface ship gas turbine propulsion unit was selected as the test-bed for this project; namely, the lubrication oil subsystem that services the reduction gears. A computer simulator of the various malfunctions affecting this test-bed was developed and tested. It incorporates normalized random variability that affects the time of triggering of alarms. The simulator operates in a manner that approximates parallel processing. Two separate diagnostic programs were constructed:

1. A computer program was developed to utilize knowledge of the order in which alarms are triggered to diagnose the cause of malfunctions. The order is represented by a dynamic tree structure.

2. A computer program, STOCHASM, was developed to utilize knowledge of the time at which alarms are triggered to diagnose the cause of malfunctions. The pattern of timing of any alarm sequence associated with a given malfunction is represented by the normal distribution of its recent values, stored in a first-in/first-out queue. Matching of the timing characteristics of unknown malfunctions to known ones is accomplished by the use of factors based upon (a) the area under the curve of the distribution of the queued samples, delimited by the time of the latest alarm, and (b) the past incidence of the current sequence of alarms.

Findings

The simulator was able to duplicate the various problems inherent in the target ship system. Use of knowledge of the sequence of alarms was found to significantly improve the ability to correctly diagnose the cause of a malfunction as it is unfolding. The methodology employed for handling knowledge of the timing of alarms worked effectively; it proved superior to utilization of only sequential information, worked well in real-time, and its learning capabilities successfully allowed it to automatically adapt to changes in the system and the environment. The degree of performance improvement was greatest early in the course of the malfunction, where it was the most useful. Possible deficiencies in the design of the target system were revealed by the program. Recommendations are made as to future requirements for fault handling of these systems.

Conclusions

The methodology for handling temporal data to enhance real-time fault detection and diagnosis is quite effective. While the program is not yet a comprehensive one, it could provide the operator with a powerful tool to assist him in deciding the cause of malfunctions as symptoms are unfolding. The methodology is generalizable to other problem domains that involve sensor-based diagnosis in real-time under conditions of uncertainty.

CONTENTS

LIST OF FIGURES

INTRODUCTION

Problem

> The progress of rivers to the sea is not as rapid as that of man to error.
>
> ... every effect clearly has its cause, going back from cause to cause in the abyss of eternity; but every cause has not its effect going forward to the end of the centuries.
>
> Voltaire, 1764.

Personnel who operate large, complex real-time systems often encounter major difficulties in attempting to cope with system malfunctions. Those difficulties are partly due to the intrinsic nature of the systems.

1. Large numbers of computations and decisions must be performed in short periods of time, and require the consideration of vast amounts of stored data. Decisions based upon the data must often be made without hesitation, yet those decisions may be critically important for system survival and are sometimes irreversible. Some of the data relevant to making those decisions may be missing. On the other hand, data that are available may be incorrect, approximate, or irrelevant.

2. For many reasons, the system targeted for control may behave erratically. Subcomponents of the system may influence one another in a complicated, nonlinear manner or may interconnect with other systems that are separately controlled in unknown or unpredictable ways. Moreover, the system may have to be operated continuously, without interruption, over long periods of time, during which the system may change its fundamental behavior as the result of numerous random, or otherwise unanticipated, factors. Even more difficult to cope with is the fact that the design of the target system (in the case of large systems) often contains a number of undetected or uncorrected flaws, so that the behavior of the system may not always correspond to predicted patterns.

In addition to operational difficulties that stem from the intrinsic nature of the target systems, there are those that result from the fact that personnel are poorly suited to the handling of many of the tasks that system designers demand they perform.

1. Humans have numerous physical limitations that restrict their performance. For example, they have insufficient computational speed to handle the thousands, sometimes millions, of sequential operations per second required to assess the state of the system. Human memory is too slow, too small, and too volatile to cope with the huge amount of data recall that is required. They are subject to many kinds of stress in operating these systems, in reaction to which their performance is likely to degrade. Furthermore, human bodies are physically unsuited to some control system environments; examples of this include possible exposure to very high or low temperatures, high radiation levels, and infectious agents.

2. Human operators vary considerably in their capabilities; optimum matches between the operator's capabilities and his assigned task are exceptions to the rule. This applies to the problem areas of personnel selection, training, motivation, assignment, genetic endowment, and availability.

Although effective manual control of such systems is inherently difficult, operators usually (and surprisingly) perform very well, provided normal operating conditions prevail. Operator failure, and subsequent system failure, is more likely to occur following a system malfunction. This is because, in the presence of a malfunction, decision-making must be performed under very adverse conditions in which the operational requirements described above are extreme in nature and often far exceed the capabilities of human operators.

An example will be cited at this point to (1) underscore the nature and extent of the problem, (2) highlight the areas in which improved techniques are needed, and (3) provide justification for the methodology presented in this report as an alternative to currently used techniques in fault diagnosis.

Nuclear Power Plant Control

The Three-Mile Island Nuclear Power Plant incident in 1979 exemplifies the inadequacies of current methods of fault handling. Nuclear power plant control systems depend heavily upon sensor-based, human-operator mediated, fault handling. The sequence of events leading to the partial meltdown that took place at Three-Mile Island has been well documented (Lewis, 1980). Briefly, the scenario was as follows:

1. Main feedwater pumps that enable cooling of the reactor failed. In response, the feedwater backup pump system activated, but the backup pump system was blocked. The blockage was caused by closed valves that were supposed to have been opened manually, immediately following the previous routine maintenance procedure; but operators had failed to do so. At this point, operator console indicator lights implied the valves were open, even though, in fact, they were not. The discrepancy was due to the lack of limit switches on the valves that could indicate their actual (as opposed to intended) state. After considerable delay, the discrepancy was discovered; the valves were noted to be closed, at which point they were opened by operators, but, by then, elevated temperatures had led to elevated steam pressure in the primary system.

2. Because of the elevated steam pressure, the reactor shutdown procedure (SCRAM) activated. Also, the emergency relief valve (to let off excessive steam pressure) opened, causing a drop in the pressure, as it was designed to do. However, after the pressure normalized, the emergency relief valve failed to return to a closed position as it should have done. Therefore, water (steam) continued to escape. That relief valve had previously been known by personnel to be defective, but had not been replaced or repaired.

3. Since coolant water was being lost through the relief valve, the temperature again rose to abnormal levels correctly causing automatic activation of the emergency coolant system. Operators responding to the automatic coolant system activation manually shut it down after deciding its activation was an incorrect automation decision. The operators assumed the problem with the plant was that it contained too much water, rather than not enough, as was actually the case.

4. The reactor core continued to overheat as coolant water continued to escape. Hydrogen began to form. Two-and-one-half hours after onset of the malfunctions, operators finally realized the relief valve was defective and still open. It was then blocked, too late to prevent damage to the reactor core.

Observations

General factors contributing to the Three-Mile Island incident can be categorized as follows: (1) defects in plant design, (2) poor quality control (construction) and poor maintenance, (3) poor management, and (4) human operational error (in the design and execution of fault diagnosis). Sufficient information has appeared in news reports to suggest that the Soviet Chernobyl incident in April 1986 involved all of the very same factors.

Certain observations concerning these incidents are inescapable:

1. Serious malfunctions can, and do, occur despite extraordinary measures.

2. Multiple simultaneous malfunctions (and/or cascading malfunctions) can, and do, occur despite repeated statements by occasional officials, researchers, engineers, or designers that this is highly unlikely (Scarl, Jamieson, & Delaune, 1985).

3. In systems of this type, humans perform poorly in the role of fault diagnosis. Despite our well-justified pride in our personnel, errors are the rule and not the exception under conditions of real-time fault handling.

Notable in most systems of this sort, great emphasis has been placed on the need to construct a perfectly fault-free system. Comparatively, much less effort has gone to the development of better techniques for fault handling. When a disaster eventually occurs, as it invariably does, the focus is thereafter directed toward a search to discover what went wrong, who was responsible, and what can be done to prevent its recurrence. But, it is unrealistic and dangerous to design and operate systems on the assumption that serious, but potentially survivable, malfunctions and mishaps will never occur. Furthermore, it is quite probable that discoveries made in the course of analyzing systems during the design of fault diagnosis programs will lead to the recognition of defects in the target system and, hence, to the more nearly perfect systems we seek to achieve. It is important to look more closely at complex and critical systems from a very different point of view: What measures can be taken to anticipate the occurrence of malfunctions or mishaps, and cope with any that might occur?

The Root Problem: Multi-sensor Integration

In process control systems generally, and in nuclear power plant control particularly, the detection and diagnosis of faults depend upon two mechanisms:

1. Large numbers of sensors are located at key points in the plant. The raw parameter values transmitted from these sensors are monitored and compared against pre-specified upper and lower range-limits of normal. Parameter values (and their designated sensor names) are brought to the attention of the operator when their range-limits are exceeded.

2. Human operators are responsible for performing (manually and in real-time) the multisensor integration (i.e., the process of observing all the sensor data, particularly the alarms, analyzing their significance, and correctly formulating a diagnosis).

Herein lies the root of a serious problem in dealing with fault diagnosis.

Design engineers and, to a lesser extent, plant operators can anticipate, reasonably well, the pattern of alarms that will be triggered by a known malfunction. To a lesser degree, this is the case even for multiple, simultaneous malfunctions.

On the other hand, they have great difficulty in reasoning in the reverse direction; that is, mapping from a complicated pattern of sensor alarms back to causative faults. Since effective methods for doing this in large, complex systems are not presently known even to the engineers who design the system, the engineers can neither incorporate fault diagnosis into the system nor provide the operator with the methodology for doing this manually. Although human operators are "responsible" for real-time multi-sensor integration, the task is often not possible for them to perform.

The obstacles encountered by operators in attempting to perform sensor-based fault handling have been widely publicized (MPR Associates, Inc. 1985; Roscoe & Weston, 1986; Seminara & Eckert, 1980; Sheridan, 1981;). These include:

1. Range-limits and Context-dependency. The range of limits of normal for any particular measured variable is context-dependent: What is normal in certain conditions may be abnormal in others. Often, instead of dealing with the need to consider context-dependency, inflexible range-limits for variables are simply empirically set at extra wide intervals so that false-positive alarms are avoided (Sheridan, 1981).

2. Alarm Subsets. Alarms, and even groups of alarms, are not necessarily uniquely identifiable with a single subsystem: The same malfunction may activate different subsets of alarms on different occasions ("fan-out"), and, conversely, different malfunctions may activate the same set of alarms ("fan-in"). Moreover, all too often, following a malfunction all possible alarms are triggered, almost at once, rather than a subset (the "Christmas tree affect") (Chambers & Nagel, 1985; Fortin, Rooney, & Bristol, 1983).

3. Priorities. A key alarm during a particular malfunction may have been assigned a low priority because it does not ordinarily have much significance. Sometimes, however, that alarm may have great significance. Therefore, the assignment of priorities and the visibility of specific alarms to an operator should be based upon the total, current operational state of the system, since priorities are context-dependent. This is not done in practice.

4. Uncertainty. The set of sensors installed into a system may not (a) supply optimal kinds of information (relevancy), (b) constitute a complete set of information necessary for diagnosis (sufficiency), and (c) transmit accurately or rapidly enough (reliability). In real-life situations, as previously described, even the domain is subject to fluctuation and uncertainty. Particularly important here is the tendency for continuous temporal changes involving almost all the variables. Incomplete system design is another form of uncertainty; unanticipated events have a nasty habit of cropping up at critical times.

5. Multiple Simultaneous Faults. Operators have no way of determining whether the alarms and their current parameter values represent single or multiple lesions. Not only do multiple simultaneous faults occur, but single faults often cascade into a series of self-perpetuating lesions ("fault propagation").

6. Man-Machine Interface. The number of alarms or relevant data is often more than can be realistically displayed, inspected, and analyzed by the operator at his console.

7. False Positives. False positive alarms are common. They may be caused either by inadequate control system designs or by defective sensors or consoles. They seriously complicate attempts to perform fault handling.

8. Real-time Constraints. Multi-sensor integration is meaningless unless done in real-time. In many of these systems, decisions must be made within seconds. There is insufficient time for the operator to manually peruse and pre-process the enormous data base involved.

The obstacles enumerated above are not meant to be exhaustive. Even so, the types of problems they embody serve to emphasize that better methods of personnel operator selection, training, testing, and motivating cannot alone make sensor-driven operator-mediated fault detection and diagnosis feasible.

Alternatives

There have been attempts to supplement or substitute for the operator in order to better deal with the problems of fault handling. They involve the use of: (1) component redundancy and voting schemes, (2) computer AI/expert systems, (3) mathematical (computer) simulations, and (4) physical analog devices. Each of these embodies a number of advantages and disadvantages. A discussion of these approaches is beyond the scope of this report, but suffice it to claim the following: When used alone, or in combinations, they do not sufficiently overcome the previously enumerated obstacles to fault handling. Even worse, these approaches, themselves, tend to introduce new, additional, and misleading uncertainties into the system.

A CONTRIBUTION TO THE SOLUTION

> All certainty which does not consist in mathematical demonstration is nothing more than the highest probability; there is no other historical certainty.
>
> Voltaire, 1764.

Objective: An Effective Method for Dealing with the Stochastic Problem

As noted previously, there are many obstacles to the overall goal of fault detection and diagnosis. One of the most difficult obstacles has been the inability to profitably make use of the timing of events that follow a malfunction in order to improve our real-time predictions of its most likely cause. We refer to this as the "stochastic problem." A methodology will be presented that can "effectively" cope with the stochastic problem. By the term "effective," we refer to the following characteristics:

1. Variability. The method must be able to work well even though all changes that take place in the system are subject to random variability (i.e., a stochastic process). Therefore, the precise time of occurrence of each new event is unpredictable. In fact, even when the very same malfunction repeats itself and happens to produce the identically same ordered sequence of events, each new event will always be somewhat different from its counterpart in the previous occurrence of that malfunction, both in its absolute and relative time of occurrence. This is because time is represented by a continuous variable whose value is expressed as a floating point number (i.e., an infinite decimal). Since the time of occurrence of any two events will, in practice, never be exactly the same, the method used for fault detection and diagnosis cannot simply

compare the time of occurrence of two events and decide whether or not they match perfectly, but, instead, must compare the times of many pairs of events and decide upon the best partial match of all of them. In short, the method must be able to cope with never-before-seen situations (data) and do so without succumbing to a combinatorial explosion in terms of the requirement for computational speed, storage, and recall of previous data.

2. Uncertainty. The method must be sufficiently robust to handle the task of predicting the cause of the malfunction despite the possibility of the phenomena of fan-in and fan-out. The diagnostic predictions must be made while the malfunction is in the process of evolving, and not simply await the full development of all eventual symptoms of the malfunction. In short, the method must be able to cope with the uncertainty of insufficient data.

3. Adaptability. The method must be able to adapt to environmental changes that affect the type, sequence, and timing of events following a malfunction. These changes include those resulting from equipment wear-and-tear, equipment replacement, alterations in temperature, design changes, etc. Moreover, the adaptation must occur with a minimal risk of false alarms. In short, the method must be able to automatically learn and remember newly occurring associations and forget outdated or newly invalid ones in an efficient, but not erratic, manner.

4. Real-time. The method must be one that will work under real-time conditions. In most systems, this implies a fully automated decision aid.

5. Predictive Power. The method must demonstrate that it can perform more effectively than alternative fault detection and diagnosis methods that do not make use of information embodied within the timing of events. (Currently, the alternative systems depend, for their predictive power, only upon the detection of specific subsets of alarms, or, in some cases, the detection of some pre-designated specifically-ordered sequences of alarms.)

6. Generalizability. Ideally, the method should be one that can be generalized to be applicable to other information overload real-time problems.

An effective methodology for dealing with temporal data is described below. It is not envisioned as a program that will undertake corrective action, but rather as one that will assist in the rapid determination of whether or not a malfunction has occurred, and, if so, the specification of its most likely cause. Its goal is that of enhancing the operator's knowledge in those areas where help is most needed . . . rapid assessment and decision-making.

No claim is made as to having invented a drastically new and entirely different kind of computer algorithm as the basis for this methodology; quite the contrary, the aim is to convince the reader that relatively simple and well-known techniques can be beneficially brought to bear on this complicated set of problems.

First, the test-bed for this program will be described; it is the lubrication oil subsystem servicing the reduction gears of the gas turbine engine propulsion unit of a Navy surface ship, the DD 963 class of destroyers. Then the methodology will be described and some aspects of its performance will be analyzed.

The Test-bed

Our selection of this test-bed was not because we believe it to be the most challenging one we could have selected, or the one in most need of this kind of application. Rather, it was selected because it is a convenient one having within it the ingredients typical of the general class of problems of interest to us. It will be readily evident that it closely corresponds to the problems of fault detection and diagnosis in space shuttle-crafts, in process control plants, and in nuclear power plants.

Description of the Lube-oil Subsystem

The gas turbine engine transmits power to the propeller shaft through reduction gears. These gears are large metal objects subject to enormous stress and friction. They form part of the mechanical transmission of the propulsion unit. In order to reduce friction and cool the gears, this part of the propulsion unit is contained within an oil-filled compartment. The subsystem that stores, supplies, circulates, filters, and warms or cools the oil for the compartment containing the reduction gears is referred to as the "reduction gear lube oil subsystem." Other parts of the subsystem include bearings, shafts, pipes, valves, sensors, heaters, filters, and motors. The subsystem may communicate with other lubrication oil subsystems that serve other parts of the propulsion unit.

The reduction gear lube oil subsystem is susceptible to approximately 27 different kinds of malfunctions. These include both major and minor faults. They may be due to human error, mechanical failure, sabotage, or wartime damage. Ship and personnel survival greatly depend upon functional integrity of this subsystem.

When a malfunction occurs involving the reduction gear lube oil subsystem, irreversible damage to vital bearings may occur within minutes. Correct diagnosis of the cause of the malfunction must be made within seconds and the proper response instituted immediately in order to maintain the integrity of the propulsion unit and fulfill the mission of the ship. Responses that might follow the detection of a malfunction in this system include closing a valve, running the engine full ahead in spite of the malfunction, reducing engine speed, or stopping the engine entirely. Exactly which response is appropriate depends upon the cause of the malfunction as well as upon the ship's environment and mission.

Prompt diagnosis is also necessary in order to prevent further deterioration of the system and to accomplish or arrange for timely repair.

Real-time Limitations of the Subsystem

Current operational reduction gear lube-oil subsystems do not allow real-time fault diagnosis, as evidenced by the following:

1. The "Christmas Tree" Effect. As in the case of nuclear power plants, multiple alarms may be set off simultaneously. In the lube oil subsystem, operators relate that it is often the case that all alarms are triggered at once. No information is available on the operator's console as to the sequence of the activation of alarms. The operator may, after-the-fact, request a computer printout of the alarm activation data, which, of course, consumes precious time. Unfortunately, the frequency at which the computer samples the sensors for data is too slow to obtain meaningful timing information for the printout. Experienced operators relate that, in about 45 percent of the cases, alarm printouts obtained even after full expression of all eventual symptoms resulting from a

malfunction show sequences that are not specific (i.e., the set of alarms has multiple possible causes). Worst of all, irreversible damage to propulsion bearings has already occurred (in some cases) by the time the first alarm is triggered.

2. Watch-standers. The set of alarms and sensors is incomplete. Missing sensor information is supplied by watch-standers. These are personnel whose job it is to visually inspect the dials, gauges, and integrity of the peripheral equipment. These objects are often located in cramped, dark, inaccessible, and potentially dangerous bowels of the ship. They cannot be inspected simultaneously or quickly.

3. Technical Manuals. Operators rely largely upon technical manuals (such as the "Engineering Operational Sequencing System") to look up and obtain guidance in making a diagnosis and instituting action. The manuals are incomplete, contain errors, and make no attempt to cope with the mutual overlapping of symptoms and malfunctions (fan-in and fan-out). The manuals do not correspond with the real-life situation, where the operator must deduce the cause of a malfunction, starting with the knowledge of which alarms have thus far been triggered (variously referred to as the "bottom-up," "data-driven," "sensor-based," or "forward-chaining" method of diagnosis). Instead, they are arranged so that if the operator knows the cause of the malfunction (which is not the case) he can then locate a list of some of the alarms or symptoms that might possibly result and match that list with the set of alarms already triggered (this could be helpful only if there were no fan-in, where many different malfunctions can set off the same set of alarms; again, this is not the case).

4. Stress and Time. The decision process may be a very stressful one, particularly when the ship is performing a critical maneuver in high seas close to other ships, or is in a wartime engagement involving enemy torpedoes or missiles. Possible complications of a malfunction include the loss of ship power and maneuverability, oil fires, explosions, or disintegrations of engine components that are revolving at high speeds (early in the case of the Chernobyl incident, it was rumored that gas turbine engine fan blades had disintegrated and subsequently penetrated the nuclear reactor, precipitating the disaster), and death of operating personnel. There may be insufficient time for the operator to arrive at a critical diagnosis and decide upon corrective actions.

5. Diagnostic Steps as a Cause of Damage. Often, the engine must be restarted after a shutdown simply to make a diagnosis of the malfunction by prolonged operational observation. This, in turn, causes further engine damage. Once damage has occurred, repair to an engine may take several weeks or more, during which time the ship is disabled.

6. Complexity. The lube-oil subsystem is a relatively small one. One indication of this (not always a reliable indicator, incidentally) is that it utilizes a maximum of only 18 console alarms. Nonetheless, real-time diagnosis of the cause of a lube-oil subsystem malfunction is difficult because of the functional and anatomical interconnectivity of its components. Any defective component quickly and adversely affects the function, and eventually the integrity, of others through these multiple and cyclic communications. Figure 1 illustrates this.

Clearly, there is need for a new and better approach to cope with malfunctions in this subsystem. Furthermore, there are indications that an approach using only the additional information of the order in which alarms are triggered will not suffice because of persisting ambiguity. This test-bed is, therefore, considered to be a good one to assess the additional value of an approach making use of temporal data.

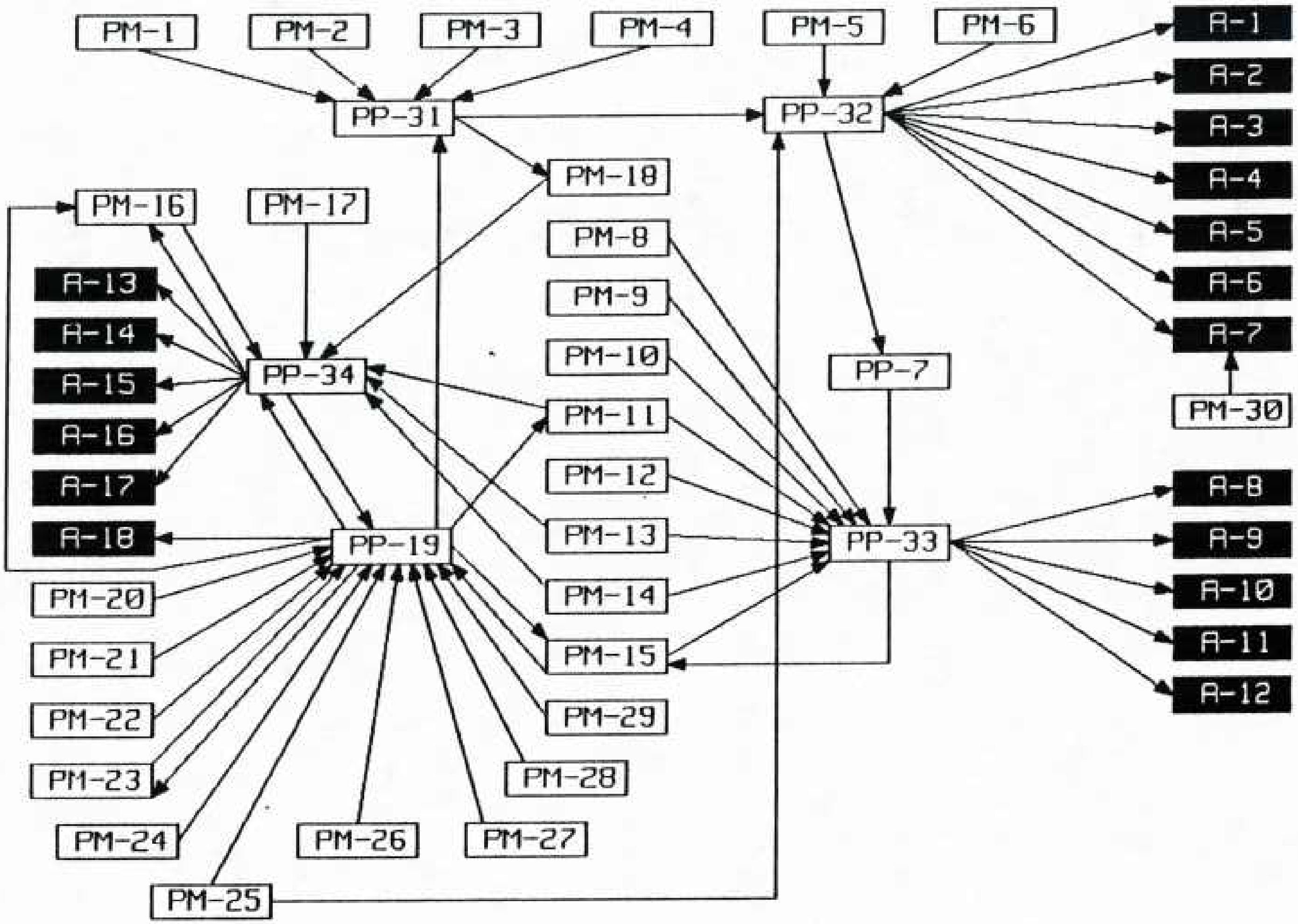

Figure 1. Reduction gear lube-oil subsystem.

A New Approach

Description of the Computer Program "STOCHASM"

A computer program that embodies a methodology capable of effectively handling temporal data will now be described. It is referred to as "STOCHASM." The program is composed of the following parts:

1. LUBE-OIL. A simulator of the behavior of the physical plant following the onset of a malfunction. More specifically, it simulates the changes in value of the parameters monitored by the reduction gear lube-oil subsystem sensors.

2. WATCH-STANDER. A monitor of the outputs of the simulator used to detect the occurrence of parameter values indicating actual or imminent triggering of alarms.

3. PATTERNS. A pattern recognition unit that maps and matches sequences of alarms, from evolving malfunctions, into stored data structures.

4. DECISIONS. A diagnostic routine that utilizes temporal data to complete the pattern matching.

LUBE-OIL--The Simulator Portion

Figure 1 was used as the basic framework for construction of a computer simulator of the malfunctioning lube-oil subsystem. Each node, or box, in the graph represents one of the following:

1. A subsystem malfunction (Pm) that can set off the whole chain of events. For example, malfunction Pm-1 represents the rupture of a main pipe transporting lube oil from one part of the subsystem to another.

2. An alarm (A) that can be triggered by the effect of a malfunction. For example, alarm A-15 indicates that the temperature of some of the propeller shaft bearings has exceeded the upper limits of normal.

3. An abnormal intermediate condition (Pp), or system state, that can eventually develop as the result of a malfunction and can go on to cause other malfunctions, alarms, or abnormal conditions. For example, state Pp-32 represents a loss of subsystem lube oil pressure that can result from:

   a. failure of a pump (Pm-6), from any one of several possible causes,

   b. obstruction to flow (Pm-5), from any one of several possible causes, or

   c. a major leakage of oil from the subsystem (Pp-31), again, from any one of several possible causes. Eventually, this loss of pressure will directly trigger a number of alarms (A-1 .. A-7), but will also lead to increased friction and temperature of bearings (Pp-33), then to surface erosion of the bearings (Pm-15), causing vibrations of the gears and/or shafts (Pp-19), and so on.

There is, of course, a very close correspondence between these boxes in the simulator and specific physical parts of the actual reduction-gear lube oil subsystem. The arrows (directed arcs) that connect two boxes indicate that the occurrence, or activation, of the condition represented by the box at the origin of the arrow can lead to the development, or activation, of the condition represented by the box at the destination of the arrow.

Within the simulator program software, each box is represented as a record ("frame") in which is recorded all the information that determines (1) how the condition represented by the box will affect other conditions or alarms, and (2) how the other conditions are permitted to affect this box. This information includes things like:

1. The current state or value of the process or alarm. For example, if the alarm represents temperature, what is its current value, what are the upper and lower range-limits of normal, and has the alarm been triggered yet? If the box represents a condition of low oil pressure, has it reached a symptomatic level yet, is it affecting any other processes in the subsystem, and, if so, to what degree? What is the threshold for symptomatic activation?

2. The manner, rate, and extent to which boxes can influence one another. For example: Is there a time delay before the temperature begins to rise? At what rate does it change? What is the pattern of its change (i.e., does it increase in value according to a linear, logarithmic, or a sine/cosine curve)? How do several different simultaneous conditions combine their effects upon a common target?

The simulation of a specific malfunction is begun by changing that box to indicate activation has occurred, and then starting to run the simulator. The simulator thereafter runs repeatedly, over and over again, until the malfunction eventually has triggered all 18 possible alarms (this happens in all cases). Each run, or cycle, of the simulator corresponds to the passage of an arbitrary unit, or interval, of time. Exactly how much time elapses can be determined by the programmer. The impact of this sampling time will be discussed below.

During each cycle, the simulator evaluates all of the boxes and computes new values for all of the information (state factors or attributes) in the box as though these evaluations and computations were done in parallel. In particular, it assesses the effect of all boxes upon each other and determines whether or not the activation of a new condition or alarm has occurred. Following every computation of a new value, the value is then modified by subjecting it to a normalized, random degree of increment or decrement, so as to mimic the variability that takes place in the real world. The actual range of variability results in a change of anywhere from +5 to -5 percent. This causes substantial differences from simulation to simulation regarding the exact time at which any of the alarms exceeds its normal range, and, hence, leads to possibly great variations in the order of alarms triggered by the same malfunction.

WATCH-STANDER--The Monitor Portion

At the conclusion of each simulator cycle, all the alarm boxes are evaluated by the monitor (WATCH-STANDER). Rudimentary trend analysis is performed. The monitor checks both the current parameter values and also their projected values (trend analysis) against the preset upper- and lower-limits of normal. As soon as any check indicates that a preset limit of any alarm has been exceeded, the monitor outputs the following information to either PATTERNS or DECISIONS: (1) the number (name) of the alarm involved in this new event, (2) whether the event detected constitutes a warning (that results from trend analysis) or the actual triggering of the alarm, and (3) the time at which this new event has occurred.

PATTERNS--The Pattern Recognition Portion

Each alarm received by PATTERNS is represented by a node (record, or frame) inserted into a software binary tree data structure that constitutes the program's memory. The level (depth) of the node in the tree represents the numerical position of the alarm within this particular ordered sequence of alarms stemming from this instance of the malfunction. For example, if the alarm is the third one to have been triggered during this simulation, then the node for this alarm is at depth = 3 in the tree. Siblings of the node (additional nodes situated at the same level in that same part of the tree) represent different alarms that have been known to occur in previous sequences having identical patterns up to, but not including, this point in this sequence. Nodes contain records of the number of times this exact (partial) pattern has been previously experienced, as well as a record of the previous malfunctions that are known to have caused this particular partial pattern. Terminal (bottom-most) nodes of the tree represent the last of the 18 alarms to be triggered during a malfunction.

For example, if the first instance of malfunction "X" were to begin with the sequence of alarms "a," "b," "c," and "d," and the second instance of "X" were to begin with the sequence "a," "b," "d," and "c," the tree would be represented by Figure 2.

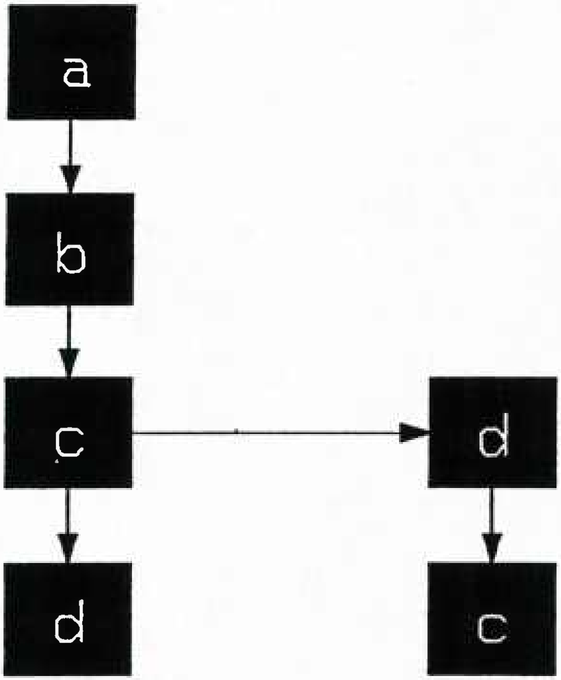

Figure 2. Tree representing early stage of a typical malfunction.

The triggering-time of a particular alarm in a particular sequence is inserted into, and saved, in a circular, first-in/first-out queue that is linked to the alarm's node. A separate queue is established for every malfunction known to create this same partial sequence of nodes (Figure 3). If this same partial sequence, having the same cause, occurs on many occasions in the future, the time values saved in the queue will accumulate and eventually fill the queue. The most recently encountered time values will then begin replacing the oldest ones. The length of the queues could be a fixed number, but, in the more interesting case, their lengths will be variable and under the control of a higher-level supervisory routine. The supervisory routine will regulate the queue lengths on the basis of the rate at which the system is encountering never-before-seen sequences of events, the degree to which the time values deviate from previous patterns and other factors.

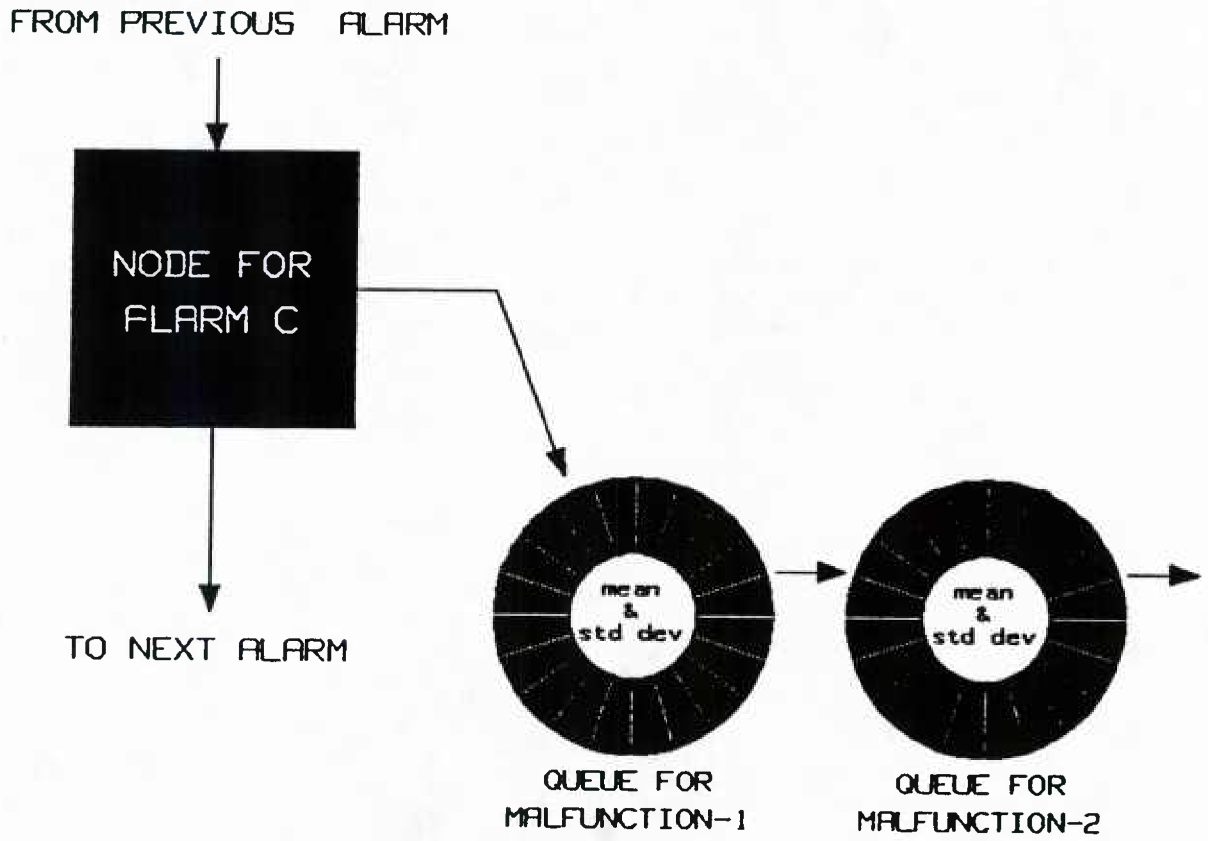

Figure 3. A tree node linked to several queues of time-values.

With each addition of a new time value to a queue, PATTERNS computes the mean and standard deviation of the curve represented by the whole collection of time values in the queue.

DECISIONS--The Diagnostic Portion

When the program is set to the task of diagnosing a newly occurring malfunction, alarm information is sent by WATCH-STANDER to DECISIONS. As each new package of alarm information is received, DECISIONS descends to the next lower level of the tree (Figure 2) and locates the node that corresponds to this specific alarm and sequence. If there is no node that corresponds with the new alarm, DECISIONS creates one and recognizes that this pattern is one that has never been previously encountered. Otherwise, DECISIONS must now determine the most likely causative malfunction, based upon the evidence (the pattern of alarms that have been triggered) so far. There may be more than one possible cause for this pattern. Each possible cause will be represented by a circular queue linked to the current node (Figure 3). For each one of those queues, DECISIONS determines how closely the new time value fits into the distribution of the time values already in the queue. The method for doing so is as follows:

1. It is assumed that the time values in the queue have a normal distribution (Figure 4). A Z-Value for the new alarm is determined on the basis of the mean and standard deviation for the normal distribution curve of the queue values and the time value of the new alarm. For example, if the mean is M, the standard deviation is S, and the new time value is T, then the Z-Value is computed as follows:

$$Z = (T-M)/S$$

This is done for each of the queues associated with the current node. Using a standard table, the Z-Value determines the area under the normal curve representing the probability that any random variable having the standard normal distribution will take on a value between that of the new alarm and the mean of the queue distribution. All of this, of course, is elementary statistics.

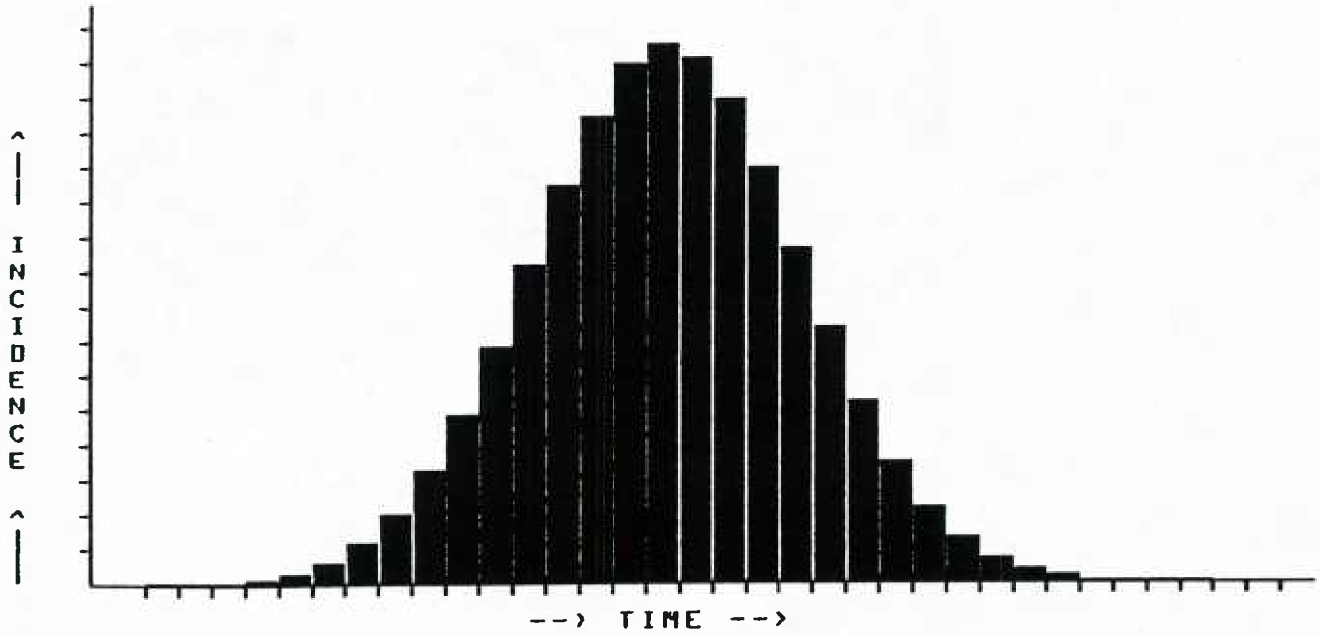

Figure 4. Actual distribution of queued trigger-times in a hypothetical case.

What we really want, however, is a probability or factor that reflects the degree to which the specific time value (T) for the new alarm is likely to belong to this particular queue distribution. That factor is defined as the magnitude of the remaining outside area ("AREA") on the same side of the mean as the new time value (refer to Figures 5 and 6 for examples of how new time values are fitted into the normally distributed set of queue values shown in Figure 4).

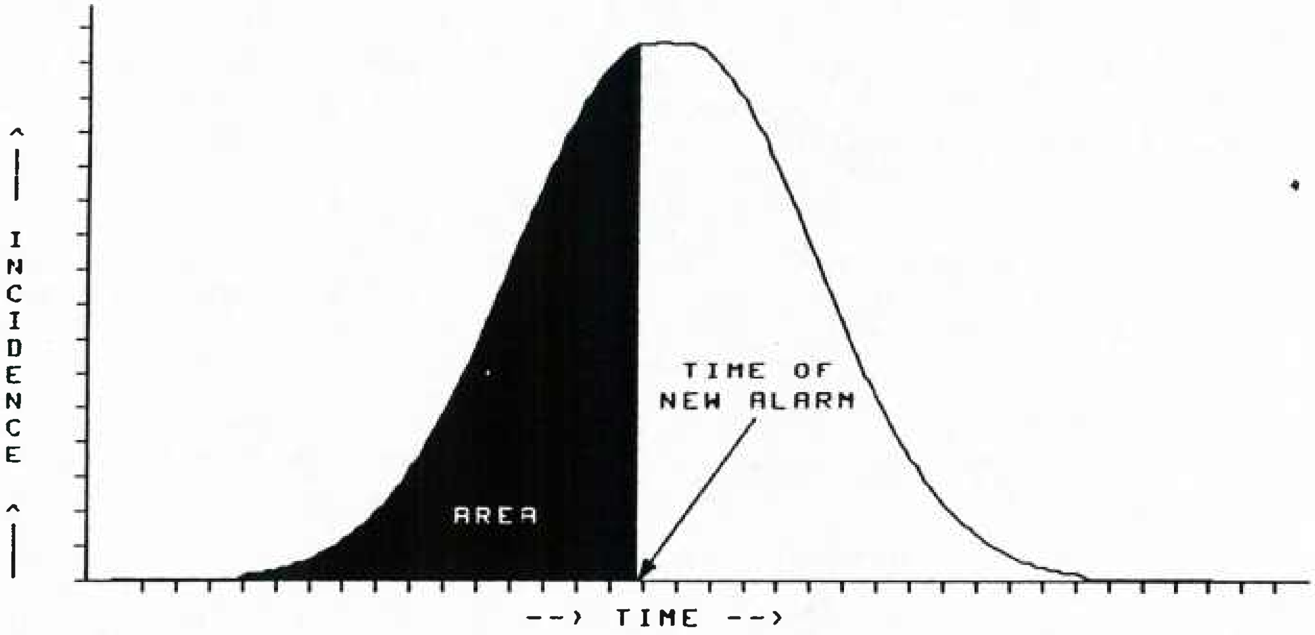

Figure 5. Example 1: Area assignment to a new alarm time.

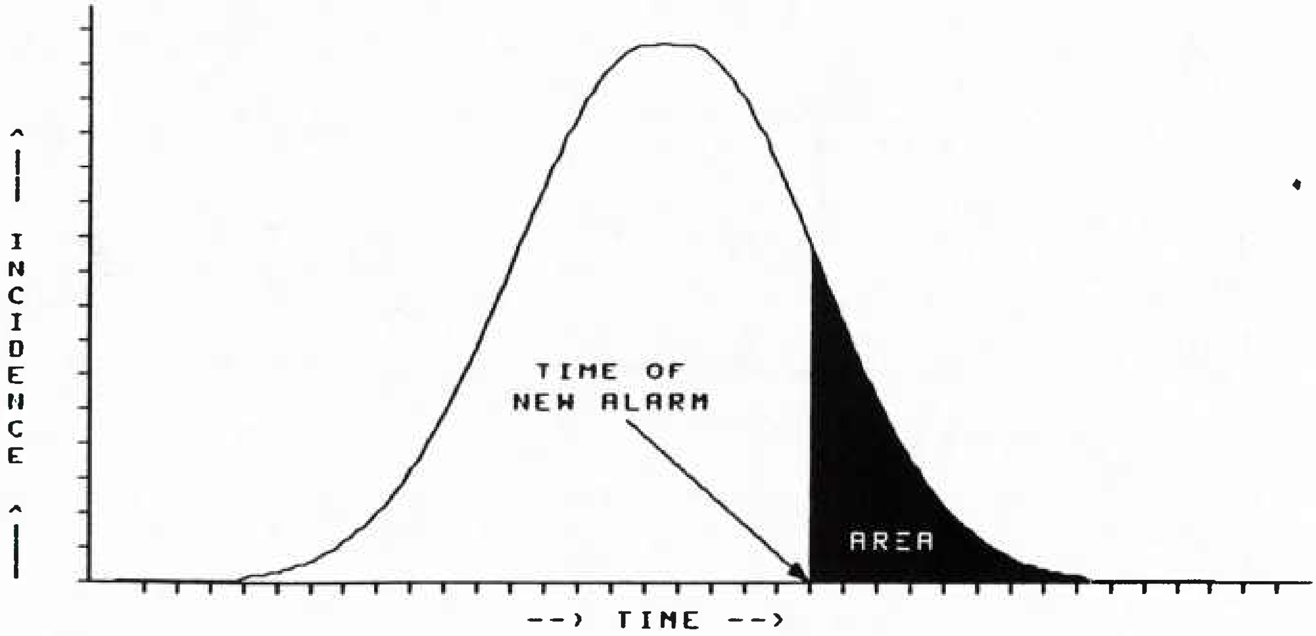

Figure 6. Example 2: Area assignment to a new alarm time.

2. The result of the previous table lookup ("AREA") is multiplied by the actual incidence of this partial sequence, yielding Pq1 (for queue number 1 corresponding to malfunction number 1), Pq2 (for queue number 2 corresponding to malfunction number 2), and so on. In effect, this means that the closeness of fit depends upon two major factors: First, the relative frequency at which this particular malfunction has been known to cause this alarm pattern in the past, and, second, the closeness of the current time value to the pattern, or distribution, of time values for this malfunction in the past.

3. The final "probability" value, or factor, for each malfunction (Pn) is determined by expressing its Pq value as a percentage of the sum of the Pq values for all the queues linked to the current alarm node:

$$P1 = (Pq1) / (Pq1+Pq2+ . . . +Pqn)$$

4. Following full development of the course of alarms resulting from a malfunction, if feedback is made available as to the actual cause of the malfunction, PATTERNS then enters the time value for each alarm in the sequence into the appropriate circular queue at each level of the tree.

While this description may sound somewhat tedious, there are, in fact, only a very small number of simple computations, or table look-ups, that need to be done, so the program runs very quickly.

Training and Testing Protocol

The Training Period. STOCHASM was run 400 times for each of the 27 possible malfunctions. At the conclusion of each run, the cause of the malfunction was made available as feedback to the program. During this period of time, DECISIONS was inactivated. The effect of this training period was to incrementally build a tree that incorporated STOCHASM's memory, or past experience, regarding the sequences of alarms (and their associated times) that result from repeated episodes of each of the malfunctions. Although the nature of the tree is altered with each new episode, the tree itself is permanently retained in memory. In effect, automatic learning is taking place in this period. STOCHASM, via DECISIONS, will, in the subsequent testing period, make use of the information stored in this memory-tree when attempting to make predictions.

The Testing Period. When the training phase was completed, DECISIONS was activated and STOCHASM was then run again through each of the 27 possible malfunctions 200 additional times. The purpose now was to check how successfully STOCHASM was able to diagnose the cause of unknown malfunctions. Data was tabulated for each instance in which an alarm was considered in a "warning" state (as the result of WATCHSTANDER's trend analysis) or had actually been triggered. The data included the:

1. Time of the alarm.

2. Time of all previous identical alarms seen by the program when the partial sequence was the same, sorted out according to the causative malfunction (data in the circular queues).

3. Alarm number and name.

4. Type of alarm (warning or triggered).

5. Actual malfunction being simulated.

6. Order of alarms thus far.

7. Degree of fan-out and fan-in at each step in each sequence.

8. Prediction of the cause of the malfunction, based upon the actual statistical incidence of all malfunctions that had been encountered at that point in the past (done for each and every partial sequence ever seen by the program).

9. Probability predictions made by DECISIONS, taking into consideration both (a) the past incidence of sequences of alarms, as done in item 8, as well as (b) temporal alarm data.

The predictions of item 8 above reflect the actual past incidence in which specific patterns were caused by specific malfunctions. They do not take into consideration temporal data, only the ordered sequence of the alarms. For example, it might determine that when the sequence of three alarms triggered so far consists of alarm number 4, followed by alarm number 7, followed by alarm number 2, then 71 percent of the time (in the past) this sequence was caused by malfunction number 16, and 29 percent of the time by malfunction number 3. Hereafter, these predictions (item 8) are referred to as the performance of the "BETTER OPERATOR," as opposed to the performance of STOCHASM, which corresponds to item 9 above.

Using the previous example for the BETTER OPERATOR, STOCHASM would attempt to further distinguish between malfunctions numbers 16 and 3 by also taking into consideration that the time intervals between alarm-pair numbers 4 and 7, and between alarm-pair numbers 7 and 2 match more closely to malfunction number 3 than to malfunction number 16.

Testing was performed repeatedly, in a similar manner as above, under a multitude of different conditions in order to empirically assess the effect of varying the:

1. Number of training runs.

2. Magnitude of random variability used in computing parameter values.

3. Length of the sampling time periods.

4. Shape of the distribution curve used in determining the random variability of time values for events.

5. Number of testing trials for each malfunction.

6. Number of previous time values retained in memory (in the circular queues) for each event.

RESULTS

The Distribution of Temporal Values

When discussing the operations of the diagnostic unit, DECISIONS, it was stated that when the same malfunction repeats itself the time values of corresponding events within identical alarm sequences are assumed to have a normal distribution, as would be expected when monitoring real-life situations. As previously described, in order to emulate this, the simulator LUBE-OIL was constructed in such a way as to incorporate into all computations a normally distributed random variability, ranging from +5 to -5 percent. Data were gathered to check how successfully this distribution was reflected in the final outputs of LUBE-OIL.

Figure 7 shows the actual distribution curve for the time values of the triggering of one of these events. In this case, a chi-square test for goodness of fit yields the value of 15.6 where $X^2_{(.95)} = 32.7$ and $X^2_{(.05)} = 11.6$. This suggests that the simulator does a fairly good job of incorporating and propagating a normally distributed variability into its computations for the changes in parameter values over time.

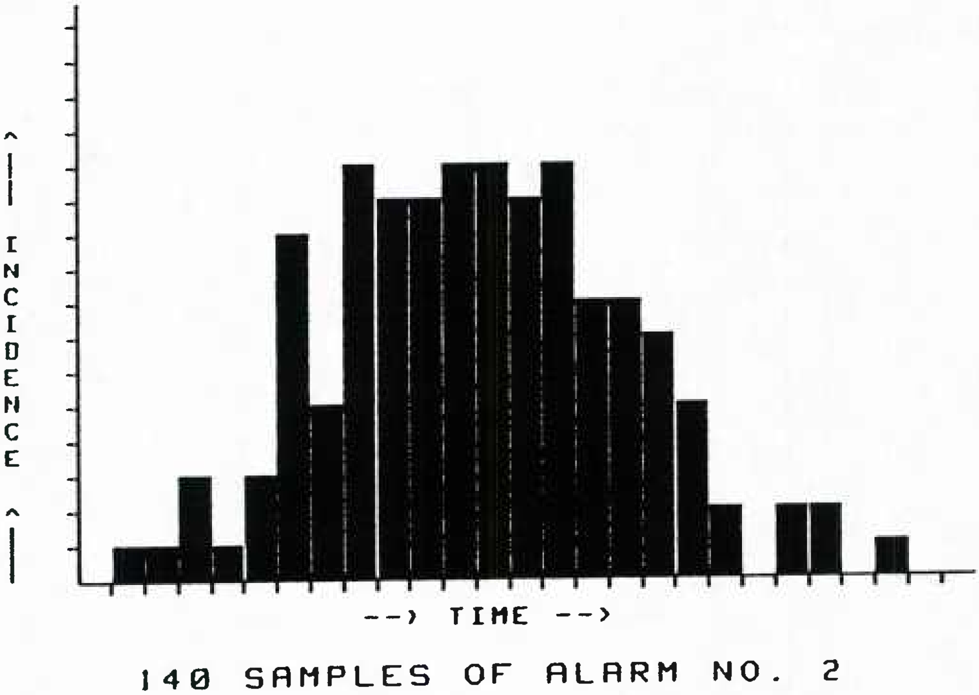

Figure 7. Distribution of trigger-times.

Fan-out

As seen in Figure 1, there is a mild, but nonetheless, significant degree of functional interconnectivity among the components of the reduction-gear lube-oil subsystem. Furthermore, the amount of time required for a change in one component to affect its neighbors is variable. Under these conditions, repeated instances of the same malfunction might be able to cause more than one unique sequence of alarms (fan-out). For example, in the training and testing runs that STOCHASM was exposed to, from all possible 27 malfunctions, there were a total of 364 different sequences of alarms noted, with each sequence made up of 36 separate events (alarm warnings or triggerings). These 364 sequences were analyzed to determine the extent of fan-out, on the average, for individual malfunctions.

Figure 8 ("FAN-OUT") plots the average number of different partial alarm sequences seen as a function of the number of alarms set off thus far during the developing malfunction. Note that the problem of fan-out is mild at first, when the malfunction just begins, but accelerates very rapidly at later stages. By the time a malfunction reaches completion (all alarms have been triggered), there are, on the average, 24 different sequences by which the malfunction may manifest itself, with a range of 4 to 38. As the figure indicates, fan-out affects all the malfunctions. By the time any malfunction is less than half developed, 100 percent of its possible causes can be shown to have previously manifested themselves by more than a single pattern of alarms.

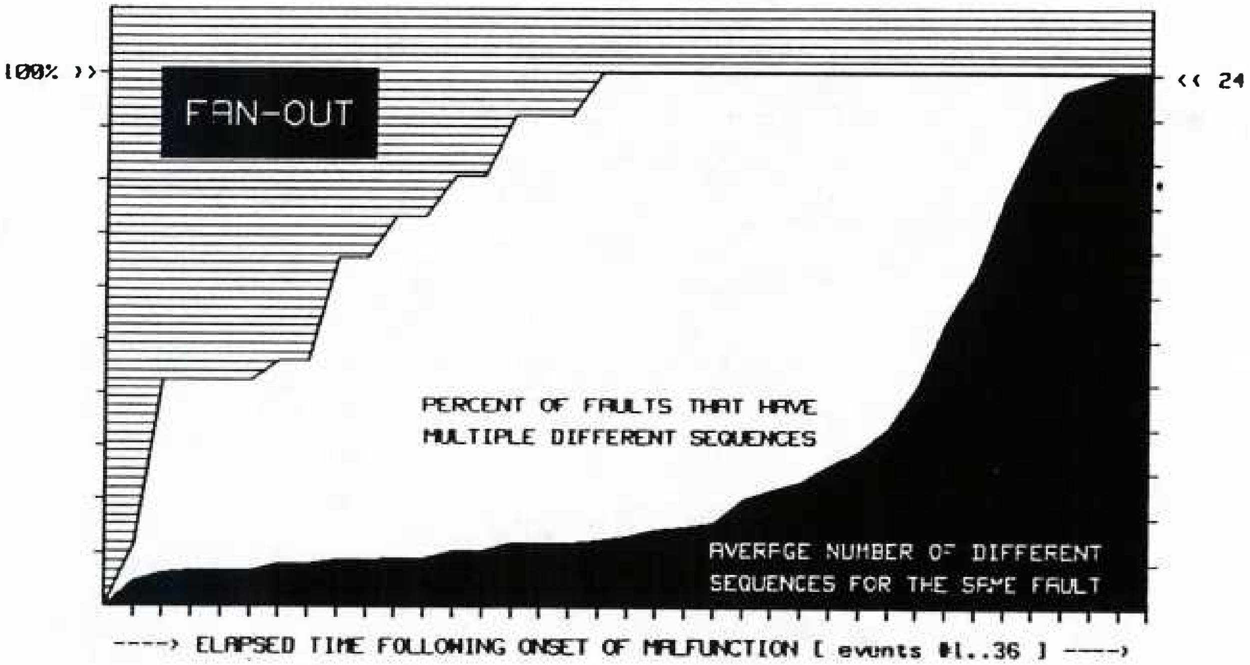

Figure 8. Degree of fan-out of alarm patterns.

From the point of view of the operator, who must perform symptom-based diagnosis, this means that his memory burden increases proportionally to the number of possible different sequences due to the same cause. He must recall each of the many different alarm sequences by which every malfunction may manifest itself. In other words, at every step of the malfunction the operator must be able to correctly answer the questions:

1. "Is this a valid, known pattern of alarms for any one of the possible malfunctions?"

2. "If so, for which one?"

3. "What has been the past incidence of this pattern as an indication of that particular malfunction?"

Fan-out, then, is one of the factors contributing to "information overload." The problem may be even worse where the different sequences are only partially different and they share the same initial subpatterns in the early portion of the sequence. In this case, the operator must not only remember the different patterns, but be able to recall the exact points in the sequences where the several patterns diverge.

Fan-in

The data in Figure 9 ("FAN-IN") serve to give an estimate of the degree of convergence, or overlapping, of sequences from different malfunctions. Obviously, if the order in which alarms occur can be absolutely identical for more than one unique causative malfunction, then the operator (or computer) cannot possibly distinguish among the possible causes on the basis of alarm order alone.

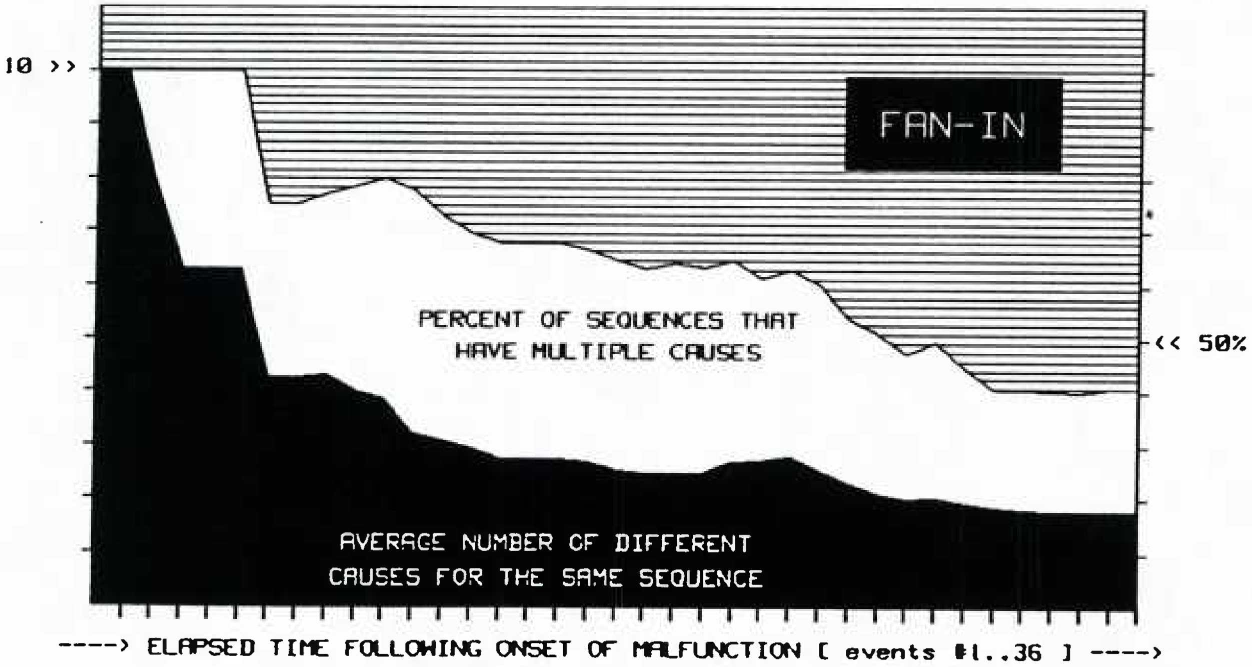

Figure 9. Degree of fan-in of alarm patterns.

At the beginning of a malfunction, as shown in Figure 9 there are, on the average, 10 different causes for each possible partial sequence of alarms. In some cases, there are as many as 15. As the malfunction develops, this reduces to an average of 2. Noteworthy is the fact that:

1. As illustrated in Figure 9, early in the course of a malfunction, there are NO sequences that uniquely determine their cause; 100 percent of the patterns have previously had multiple possible causes.

2. At the conclusion of the malfunction, 149 out of all the possible 364 sequences, or 40.9 percent, still have multiple possible causes. Compare the closeness of this figure with the one of 45 percent estimated by the operators (noted previously under the section "Real-Time Limitations of the Subsystem"). This closeness gives us some assurance that the simulator model used here is reproducing the kind and degree of problems faced by operators of real systems.

Performance of the "BETTER OPERATOR"

As defined in the previous section on methodology, the "BETTER OPERATOR" is the part of the computer program that records, updates, and has available the actual historical incidence of all possible causes for each and every partial or complete sequence of alarms. Naturally, one would expect the performance of the BETTER OPERATOR to significantly exceed that of the human operator. The human has little or no real-time information or historical data regarding the possible sequences of alarms or their significance, and so can do little better than random chance. In our evaluation of DECISIONS, we need to know not only how much better it performs compared with the human, but also how it compares with the BETTER OPERATOR who has the advantage of computer memory and speed, but lacks the advantage of temporal information.

In Figure 10 ("PROBABILITY ESTIMATE") the performance of the BETTER OPERATOR is plotted. This is averaged for all of the possible malfunctions, with each malfunction run 200 times, and subject to random (normally distributed) variability throughout. Performance is expressed in terms of the actual incidence on record in the data bank of BETTER OPERATOR regarding the likelihood that the causative malfunction is the one that is actually in the process of occurring. The incidence is updated with each new alarm event. The numerical sequence of events, from the beginning to the end of the simulated malfunction, constitutes the X-axis. For example, following the fourth event, BETTER OPERATOR, on the average, assigns an incidence (probability) of 0.169 to the particular malfunction that happened to be the actual cause of that sequence at that instance.

Note the gradual and linear slope of the plotted curve, progressing from an initial average incidence of 10.8 percent to a final value of 51.6 percent.

When BETTER OPERATOR is asked to rank the various possible causes at each point along the way, a similar curve is produced (Figure 11, "RANK ASSIGNMENT").

Performance of "DECISIONS"

The performance of DECISIONS was measured in a similar manner, using its "probability" (or "factor") predictions in place of the incidence data used by the BETTER OPERATOR. For the sake of ease of comparisons, these results are superimposed on the previous plots in Figures 10 and 11.

In contrast to the BETTER OPERATOR (where temporal data is not utilized), DECISIONS earlv and rapidly focuses upon the correct diagnosis, and, in general, consistently maintains its advantage. There is a prominent, short-lived drop-off in performance, indicated by the arrow. This will be the subject of discussion shortly.

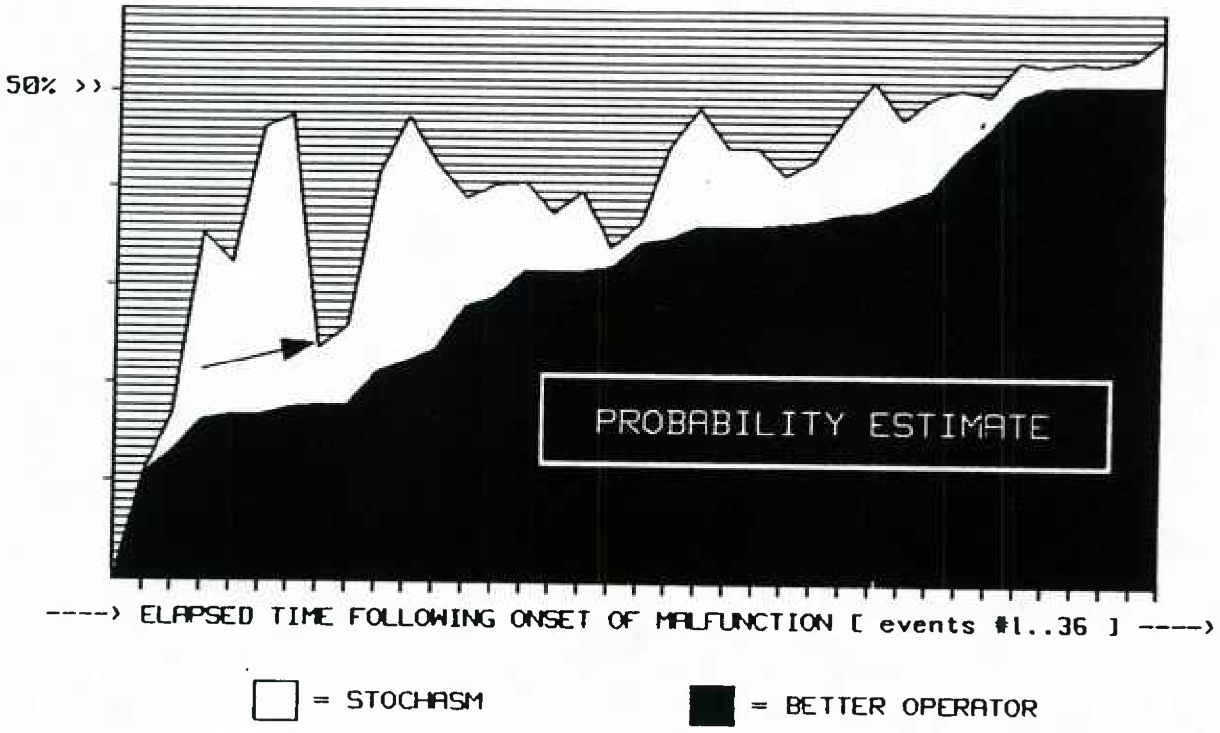

Figure 10. Performance comparisons 1: STOCHASM versus BETTER OPERATOR.

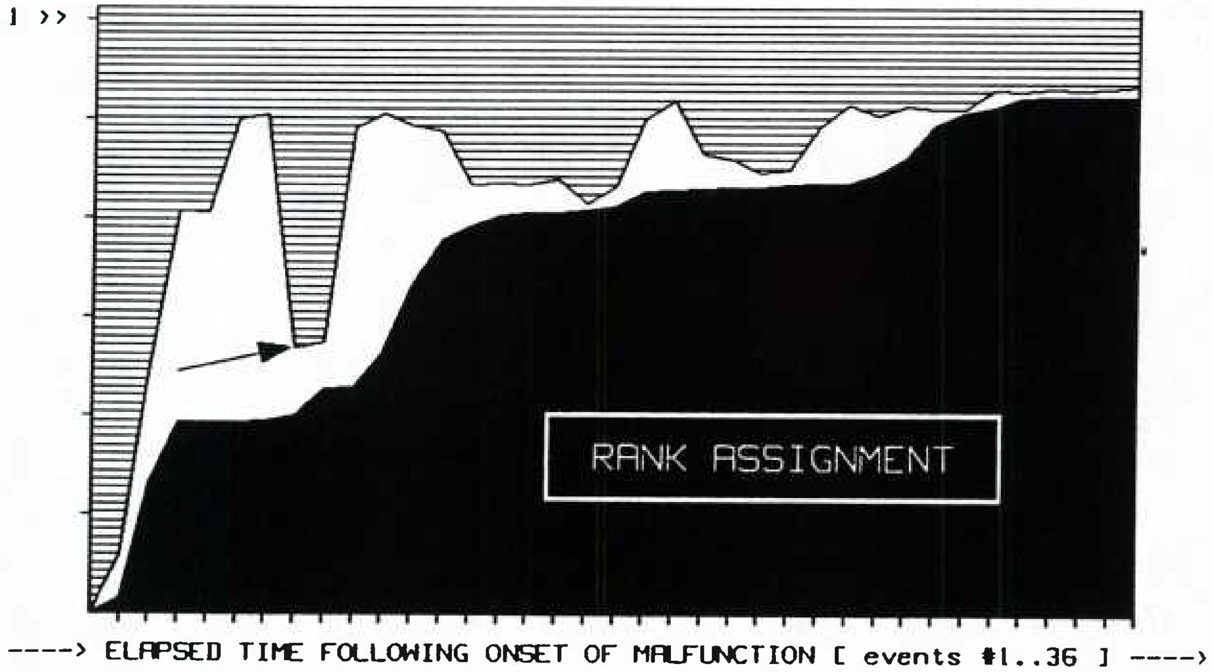

Figure 11. Performance comparisons 2: STOCHASM versus BETTER OPERATOR.

DISCUSSION

Performance

It is quite clear from the performance curves (Figures 10 and 11) that the approach using temporal data, in DECISIONS, provides a considerable diagnostic advantage. Moreover, the enhanced diagnostic power of DECISIONS is most evident during the early phase of a malfunction when it is most needed and most beneficial. Early diagnosis is essential if corrective action is to be effective and damage to be avoided; yet, early diagnosis, using the BETTER OPERATOR or the unassisted manual operator approach, is not possible in many systems (see the discussion below on fan-in/fan-out). This, of course, should not surprise anyone, since, intuitively, a program that has available to it additional important and relevant information would be expected to outperform programs that are not so well-informed. The point is:

1. DECISIONS is able to effectively use that information, despite the uncertainty of the data.

2. The method used by DECISIONS is such that the benefit derived from using this additional information far exceeds the computational costs (i.e., it is an efficient method that is well worth using).

3. DECISIONS works effectively and efficiently in a real-time environment.

4. STOCHASM is a finished product that works. While it is a fruitful "research tool" and is capable of further enhancement, as discussed in Part 1 of this report, its methodology could be applied to fault detection and diagnosis problems as a decision aid now.

Factors Affecting Performance

The performance quality of DECISIONS depends upon many factors, including the sampling period, degree of sample variability, and the type of sample distribution curve. For example:

1. If the time intervals at which LUBE-OIL cycles (and WATCH-STANDER checks the alarm boxes) are sufficiently small and the degree of variability in the time at which an event (alarm) is triggered is comparatively large, then, when a malfunction repeats, corresponding events will not significantly overlap in time (share the same time-slice value) and the spread of the time distribution curve for the values in the circular queues will be large (i.e., the curve will have a large standard deviation) (see Figure 12).

2. On the other hand, if the time intervals are sufficiently large and the degree of variability in timing is comparatively small, corresponding events will greatly overlap in timing, the standard deviation will diminish, and the samples will tend to cluster into two main groups: (a) those that are close to or identical to the mean of the distribution, and (b) those that are not (see Figure 13). Interestingly, to whatever extent overlapping occurs, the effect approximates that which is achieved by not treating time values as continuous variables, but, instead, subdividing the full range of possible time values into subranges (refer to the section below, "Comparison of STOCHASM with Other Attempts to Deal with Temporal Data" for further discussion and examples).

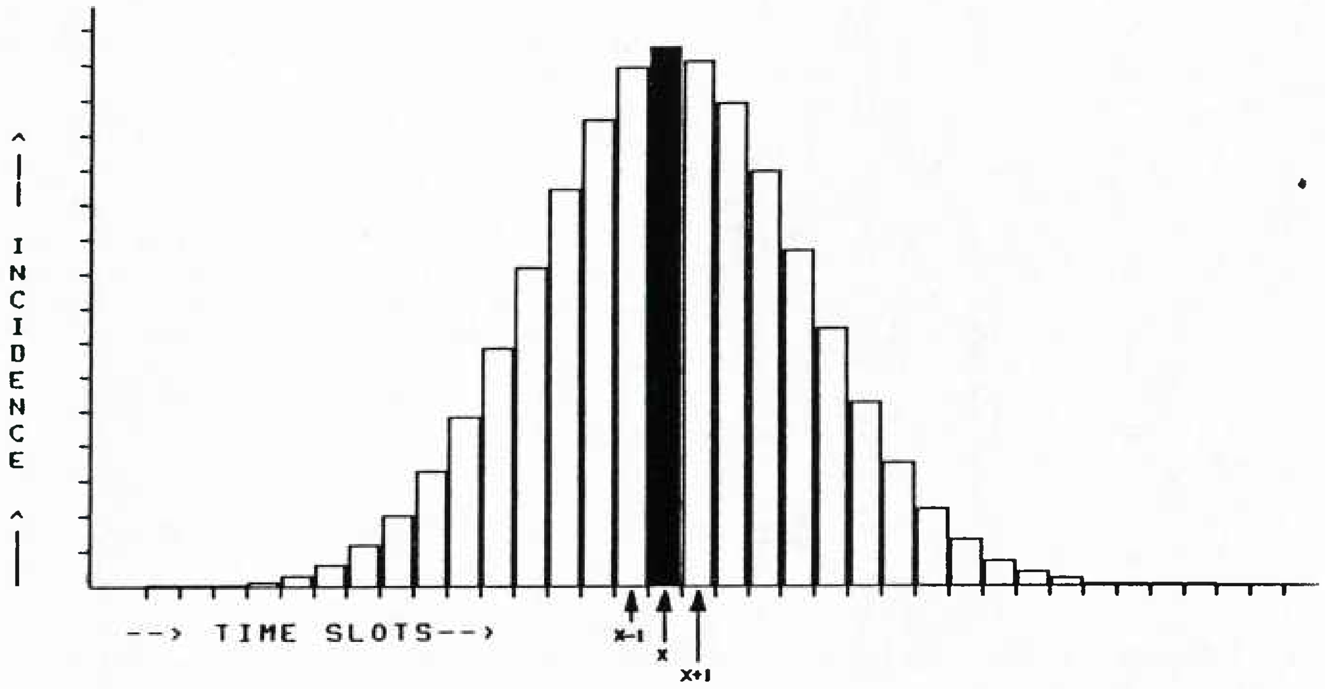

Figure 12. More frequent sampling.

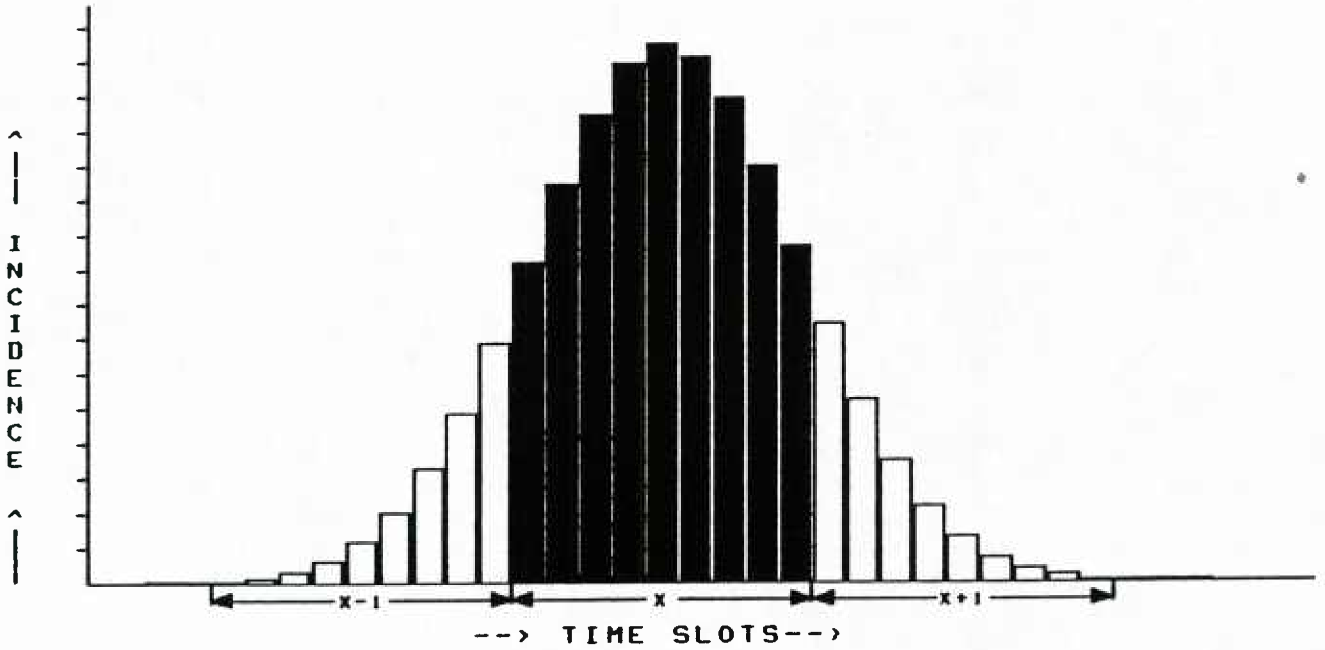

Figure 13. Less frequent sampling.

This latter effect, wherein time values tend to cluster and assume one of a small number of possible values, can be used to advantage, especially when carried to the extreme where the number and distribution of subranges causes any given new time value to either match perfectly with a specific fault or not match at all (binary). It allows the possibility of more sharply distinguishing between alternative considerations for the cause of a fault. The use of this effect in this manner results in a program that more so resembles a heuristic algorithm in performance as opposed to a probabilistic one. The end result is analogous to the use of very high contrast film in photographic printing to eliminate shades of gray. Furthermore, the presence of this effect diminishes the importance of the assumption in this domain that the distribution of time values conforms to a normal curve.

The disadvantage of increasing the sampling time relative to the degree of variability in alarm times is that this correspondingly diminishes the accuracy of the "probability" predictions and will introduce occasional errors. In its actual implementation, DECISIONS performs a weighted averaging of its probability predictions from event to event so as not to be unduly influenced by an infrequent spurious result.

Incidentally, this author has recently seen a somewhat similar use of normal distribution curves as evaluation functions in exploring search trees (Palay, 1985). The treatment was confined to theoretical issues; no use was made of sampling time heuristics and there were no applications. The work is of merit and worthy of mention since it includes a lengthy validation of this particular usage of statistical curves and areas.

Assumptions and Approximations

While DECISIONS could have been implemented with a purely statistical approach, rather than a heuristic one, this was not its goal. Rather, the goal, as previously stressed, was to find "an effective method" for improving our predictive powers regardless of whether the technique involved statistics, heuristics, or any other mechanism. In doing this, approximation techniques were employed at a number of levels. The result, in effect, is best described as a "hill-climbing" approach. For example, let us assume that the circular (first-in/first-out) queues, established by PATTERNS and used by DECISIONS to make predictions, have a fixed length of 40 slots. The approximations being made include the following:

1. The 40 most recent values are representative of the entire domain currently within reach.

2. If there is any change in the domain, it will be a partial or gradual change rather than an abrupt and drastic one.

3. The nature of the domain is that it is continuous.

There are many benefits to be gained from making these approximations. New data values may be evaluated by extrapolation onto the current samples in the queues. Also, there are tremendous savings in memory storage requirements and in the need for speed in performing computations and searches. Finally, the model of the domain becomes adaptable, and, in this sense, learning is enabled.

The Mechanism of Learning and its Role

The subject of learning mechanisms is not the focus of this report. Nonetheless, a few comments are in order to explain some of the advantages of employing such mechanisms in programs of this sort and how they generally affected the implementation. This is valuable if, for no other reason, it removes some of the mystique and suspicions often associated with the term "learning."

The use of a circular queue to represent one aspect (time) of the domain allows gradual replacement of older time-values by newer ones; the representation of that part of the domain is continually updated. A shorter queue length obviously disposes to a more rapid adaptation to changes in the domain, since each new value will have a relatively larger impact upon the total sum and the mean of the distribution of samples. On the other hand, a longer queue length is more resistant to erratic behavior, since each new value has only a small effect upon the sum and mean of the sampled domain distribution. Ideally, the length of each of the queues is under hierarchical control and is, therefore, varied according to circumstances. For example, the queue would be shortened in length if feedback begins to reveal errors in diagnosis or if all members of a series of recent values are very far from the mean.

The value of this adaptability for shipboard gas turbine systems would be tremendous. The component systems of each individual ship are constructed from parts that differ in physical makeup and functional behavior from comparable parts on other ships. Components change behavior as they undergo wear-and-tear. They change according to environmental conditions, for example, as the ship travels from a tropical climate to the arctic. They may be replaced by new and different parts when they breakdown or become obsolete. Only an adaptable, "learning" fault detection system can cope with constant and unpredictable change of this sort.

Learning is also performed by PATTERNS by associating the sequential order in which alarms are received with the specific malfunction that causes that sequence. As previously described, this is done by using a dynamic data structure composed of a tree of pointer-linked nodes. As new and different sequences are experienced, the tree is enlarged, in real-time, by the addition of new linkages from existing nodes to newly created nodes representing the new sequences. The statistical and temporal data stored in the nodes of the tree could be looked upon as analogous to reinforcing and inhibiting factors that influence the strength of the connections between the nodes. Potentially, these factors could be employed by a hierarchical control structure to perform and regulate the pruning of older and seldom used branches of the tree. This would constitute another form of learning.

Another advantage of the tree structure is that different alarm sequences starting off with an identical series of alarms can share the same nodes. This affords some degree of economy in the consumption of computer memory.

An important point is that PATTERNS learns all sequences and their associated causes automatically, and uses that knowledge effectively in its pattern recognition work. Nowhere in STOCHASM are there rules that explicitly spell out any particular sequence of alarms or explicitly associate a sequence with a particular malfunction. All of this is done automatically, in real-time, using a combination of rules on how to learn from inputs and feedback.

The Significance of Fan-in to the Learning Process

Learning is impaired in the presence of ambiguity. In presenting data about the degree of fan-in, it was pointed out that, more often than not, the same sequential pattern of alarms has several possible causes (i.e., ambiguity). This is not just a matter of information overload. Without additional kinds of information it is simply not possible for either the human operator or a computer control system to discriminate between the various causes, thereby resolving the ambiguity and making a definitive diagnosis. The implications of this deserve emphasis.

Under conditions where a task cannot possibly be performed at all, it is folly to think that improvements in personnel selection, training, evaluation, or motivation can enable the human to do that task when it is assigned to him. The operator is in desperate need of assistance here.

The point is that additional information is available, but simply not being used (or being used to advantage) by most existing fault detection and diagnosis systems; it is information often in the form of temporal data.

Resolving Ambiguities

Quite obviously, the consideration of additional data (time values) by STOCHASM substantially reduced the level of ambiguity and improved performance. Note, however, that the average performance curve contains an indentation, or dip (indicated by the arrows in Figures 10 and 11), at which point performance temporarily degrades almost to the level of the BETTER OPERATOR. When looking at similar performance curves for individual malfunctions (not shown in this report for purposes of brevity), it is evident that:

1. In some cases, degraded performance does not occur.

2. In those malfunctions where temporary degradation does occur, the malfunctions can be grouped into subsets wherein the time at which the dip occurs tends to be significantly different between subgroups, but consistently the same within members of the subgroup.

Preliminary analysis of this degradation phenomenon suggests that, in these cases, either the functional structure of the system or the placement of particular sensors is such that it is not possible to take diagnostic advantage of temporal aspects of the malfunction processes. To a large degree, a timing "bottleneck" exists; the set of times at which a certain subset of the sequence of alarms are triggered is essentially the same for a whole group of malfunctions.

If this analysis should prove to be correct, it could lead to the development of a powerful tool for improvement in the design of fault detection and diagnosis systems. Performance curves of the type presented here are easily obtained by either monitoring the actual plant or by simulation studies. The degradation dips in the curves may pinpoint irrelevant sensors, suggest better locations or types of sensors, or imply the need to redesign the plant system process flow in order to achieve more effective fault detection and diagnosis.

Comparison of STOCHASM with Other Attempts to Deal with Temporal Data

There have been attempts to use temporal data for purposes of diagnosis by subdividing the full range of possible time values into discrete subsets (Tsotsos, 1985). Consider the hypothetical simplified case wherein an elapsed time sample is classified according to its membership in one of the following range subsets:

1. Subset A. Less-than-or-equal to 0.1 seconds, or
2. Subset B. More-than 0.1 but less-than-or-equal to 0.8 seconds, or
3. Subset C. More-than 0.8 seconds.

Membership in a specific subset may then be used, via computer program production rules, as evidence that indicates or favors a particular diagnosis. There are disadvantages to this methodology:

1. First of all, it presumes accurate knowledge, in advance, of the meaningful borders between subsets. In the example above, one would have to predetermine that it is most meaningful to separate samples into those that have a time value of more-than or less-than-or-equal-to 0.1, as opposed to using some other border value such as 0.2 in place of 0.1.

2. Next, it treats the border values in a context-independent manner; they remain the same under all conditions, regardless of the current values of other parameters or significant changes in the system state.

3. Further, it allows for no differentiation between time values that are widely separated from one another yet are members of the same subset. For example, in the hypothetical case above, an event having a time value of 0.15 would not, in this scheme, be differentiated from one having a time value of 0.75. Both values reside well within the same categorical subset, yet the two events are clearly not identical and the time difference between them may be very significant as a clue to the diagnosis.

4. Finally, it does not permit flexibility for the program to adapt to dynamic changes in the border values (i.e., those changes that occur as the plant is running). Again using the previous example: Initially, use of the value of 0.1 as a border between subsets A and B may be valid in the sense of best enabling a diagnosis. But the state of the system may be subject to rapid change of the kind that shifts the best possible border value, for purposes of diagnosis, from 0.1 to 0.7. Ignoring this possibility of context-dependencies of border values will result in erroneous diagnoses.

The alarm handling methodology developed by General Electric as a part of its "Advanced Nuclear Technology Operation" program (Mott, Pugh, & Cook, 1984) directly deals with the context-dependency of temporal data. Again, time values are expressed as a small set of subranges rather than dealing with time as a truly continuous parameter; and, again, there are no provisions for changes in these subrange-limits ("patterns") once they are established. Ranges, however, are computed in advance for each of the different events in an alarm sequence. The authors deserve credit for recognizing the importance of context-dependency when using timing information and for the use of an innovative approach.

STOCHASM attempts to go further and incorporates a combined approach that:

1. Uses knowledge of the ordered sequence of alarms.

2. Uses knowledge of the past history of time values (temporal information) for every alarm, in any alarm sequence, and for any possible malfunction so as to incorporate context-dependency.

3. Directly uses the floating point value of the time of triggering an alarm instead of fitting it into a small subset of range-limits.

4. Employs a "learning" technique so that the significance of alarms and their temporal values is constantly updated by experience.

In doing so, STOCHASM makes considerable use of embedded context-dependency in regard to (1) the sequential order of alarms and (2) temporal data. The drawback of the present version of STOCHASM is that it does not also treat the value of the sensor parameter that underlies the event (such as the numerical values of temperature, fluid level, rotational speed, and pressure) as similarly having context-dependency and being a continuous variable. For example, WATCH-STANDER blindly accepts upper and lower range-limits as the mechanism for determining whether or not an alarm is to be triggered. The same approach used to handle the temporal data can be modified to handle this problem of context-dependency of sensor values. This, in fact, constitutes the next phase planned for this project and is anticipated to profoundly enhance the detection and diagnosis acumen of the program.

Toward a More General Applicability--A "Black Box" Tool

The same type of approach that was used in STOCHASM, as well as that which is contemplated for the next phase of this work, can, with some modifications, be applied to other similarly structured problems with inputs of continuous variables. Indeed, STOCHASM was set up with exactly that in mind. It incorporates a "black box" mentality, assessing the significance of a sequence of inputs having a floating point format, without regard to the input domain, and blindly (but effectively) associating patterns with diagnostic names supplied by feedback. As such, there is nothing to prevent its direct application to other domains, such as medical diagnosis, lofargram assessment in antisubmarine warfare (ASW) sonar classifications, and the military mission-control/star-wars-defense-initiative problem of multi-sensor integration. One should look upon the problem of fault detection and diagnosis as only one example of a more generic problem, in which case it is reasonable to claim that a methodology effective as a solution for one example is also potentially applicable to others in the more general domain.

"LUBE-OIL" as a Model

The simulator for this program was put together with great care. Operators were consulted at many phases of its construction. Large volumes of technical ship and engine manuals were analyzed to assure that the simulator would be realistic. Considerable performance data was collected and studied to ensure, within reason, its validity. But total, absolute realism was not the goal, nor is it necessary, important, or even relevant. All that was needed was a test-bed manifesting the general kinds of behavior typical of these malfunctioning systems and to show that the methodology of STOCHASM can be used to advantage in those kinds of systems. Indeed, in many of the studies done as part of this project, the simulator was changed away from the actual typical ship system behavior in order to provide a more severe challenge to the proposed solution methodology. While it would be important in an actual application setting to have STOCHASM begin its operation with stored data that is as close as possible to the real-world lube-oil subsystem behavior, remember that data of that sort could always be obtained:

1. From a more realistic simulator, if and when necessary.

2. By analyzing actual plant system outputs over a finite period of time.

3. Directly by STOCHASM by allowing it to monitor one or more actual shipboard plant systems (as opposed to a simulator) for a finite period of time before beginning to actually use STOCHASM as a decision aid. After all, STOCHASM in no way would modify or interfere with the ongoing system of fault detection, diagnosis, and corrective action, such as it is; and STOCHASM has the advantage that it is programmed to learn from experience.

A Comprehensive Solution

As implied in Part 1 of this report, there are many facets to the problem of automated, real-time fault detection and diagnosis. A comprehensive solution must eventually deal with all of them. Those not addressed in this phase of our project include:

1. Cluster analysis of alarms (determining subsets of the alarm sequence that can be treated coherently as one unit).

2. Temporal compression techniques and other methods for comparing temporal patterns that differ because their common cause varies in severity.

3. Enhancement of automatic trend analysis (the prediction of the diagnosis while the symptoms are in the early stages of development) by dealing with context-dependency.

4. The evaluation of real-time performance, such as by measurement of elapsed times to check speed of decision making or measurement of utilized computer memory.

5. The ability to generalize and encode complex patterns of events so as to extrapolate when perfect matches are not possible.

6. Preventing faulty human maintenance and control decisions from either occurring or, having occurred, leading to the development of malfunctions.

7. Dealing with false-alarms.

8. Achieving high-level, or general learning, as opposed to very domain-specific or type-specific learning.

9. Coping with multiple, simultaneous, independent faults.

This report, then, is not put forth as a finished product, but rather as an illustration that it is possible to deal successfully with members of this group of problems, and, therefore, a comprehensive solution is a reasonable goal.

CONCLUSIONS

Relatively simple techniques, such as those in the computer program STOCHASM, can be successfully applied to the key problems of real-time fault detection and diagnosis.

STOCHASM is of considerable value in studying this general class of problems. For example, it illustrates that it can be useful in distinguishing between operator and system tasks that are feasible and those that are not; it provides insights about design deficiencies; it provides a tool for quantifying performance in these systems; it allows for the convenient handling of uncertain and time-dependent data; and it is proof that even relatively low-level learning techniques can be effectively used and are of enormous practical value.

STOCHASM has proven to be more than just an analytical research tool. It works, and works well. The methodology could be beneficially applied to actual systems, if desired, even at this stage. It is a program that is readily amenable to further enhancements, such as dealing with the context-dependencies of variables, in real-time, to achieve superior results over the use of preset upper and lower range-limits of normals. Because of its black-box structure, the program is generalizable and could be easily applied to other domains having comparable problems involving temporal data, random variability of data, and changing domains.

We should continue to attempt to design and build perfect systems; but we should also recognize that unanticipated events will continue to frustrate those attempts. Therefore, we are advised to make equally vigorous efforts to develop more advanced techniques for real-time fault detection and diagnosis to assist the operator. The computer program herein described represents a step in that direction.

REFERENCES

Chambers, A. B., & Nagel, D. C. (November 1985). Pilots of the future: Human or computer. Computers, 18, (11), 74-87.

Fortin, D. A., Rooney, T. B., & Bristol, E. H. (19-21 September 1983). Of Christmas trees and sweaty palms. Learning systems and pattern recognition in industrial control. Proceedings of the Ninth Annual Advanced Control Conference, 49-53.

Lewis, H. W. (March 1980). The safety of fission reactors. Scientific American, 242 (3), 53-65.

Mott, J. E., Pugh, L. C., & Cook, G. J. (March 1984). Intelligent alarm handling in LMFBRs using pattern recognition matrix techniques (GEFR-00707 UC-79p). Sunnyvale, CA: General Electric Company, Advanced Nuclear Technology Operation.

MPR Associates, Inc. (December 1985). Power plant alarm systems: A survey and recommended approach for evaluating improvements (NP-4361). Washington, DC: Electric Power Research Institute.

Palay, A. J. (1985). Searching with probabilities. Marshfield, MA: Pitman Publishing, Inc.

Roscoe, B. J., & Weston, L. M. (May 1986). Human factors in annunciator/alarm systems: Annunciator Experiment Plan 1 (NRC FIN No. A1394, DOE 40-550-75, NUREG/CR-4463, SAND 85-2545). Albuquerque: Sandia National Laboratories; and Washington, DC: U.S. Nuclear Regulatory Commission.

Scarl, E. A., Jamieson, J. R., & Delaune, C. I. (July 1985). Process monitoring and fault location at the Kennedy Space Center. Sigart Newsletter, 93, 38-44.

Seminara J. L., & Eckert, S. K. (March 1980). Human factors methods for nuclear control room design: Volume 4: Human factors considerations for advanced control board design (NP-1118). Palo Alto, CA: Electric Power Research Institute.

Sheridan, T. B. (1981). Understanding human error and aiding human diagnostic behavior in nuclear power plants. In J. Rasmussen & W. B. Rouse (Eds.), Human detection and diagnosis of system failures (19-35). New York: Plenum Press, 19-35.

Tsotsos, J. K. (February 1985). Knowledge organization and its role in representation and interpretation for time-varying data: The ALVEN system. Computational Intelligence, 1, (1), 16-32.

DISTRIBUTION LIST

Deputy Assistant Secretary of the Navy (PB&DNSARC)
Chief of Naval Operations (OP-03C2)
Office of Naval Technology (Code 222)
Defense Advanced Research Projects Agency
Department of the Navy, Office of the Secretary (Research, Engineering, and Systems)
Chief of Naval Research (Code 200)
Naval Research Laboratory, Washington, DC (Code 7510)
Chief of Naval Education and Training (Code 02), (Code N-2), (Code N-5), (Code N-9)
Chief of Naval Technical Training (Code 016)
Commander, Training Command, U.S. Atlantic Fleet
Commander, Training Command, U.S. Pacific Fleet
Commander, Fleet Training Group, Pearl Harbor
Office In Charge, Engineering System School, Great Lakes
Commander, Naval Sea Systems Command (PMS 400D), (NSEA 50C), (NSEA 56Z41), (NSEA 56X1), (NSEA 56X54), (PMS 390ED), (PMS 400D41), (NSEA 003)
Commander In Chief, U.S. Atlantic Fleet
Commander In Chief, U.S. Pacific Fleet (Code 03), (PEB)
Commander, Naval Surface Force, U.S. Atlantic Fleet
Commander, Naval Surface Force, U.S. Pacific Fleet
Commander, Space and Naval Warfare System Command (SPAWAR 00)
Commander, Naval Air Force, U.S. Atlantic Fleet
Commander, Naval Air Force, U.S. Pacific Fleet
President, Naval War College (Code E-1121)
Superintendent, Naval Postgraduate School
U.S. Nuclear Regulatory Commission, Division of Risk Analysis and Operations
Dr. Ackles, Canadian Defense Liaison Staff, Washington, DC
Canadian Forces Personnel, Applied Research Unit, Canada
National Defense Headquarters, Directorate of Marine and Electrical Engineering, Canada
Ministry of Defense, Senior Psychologist, England (Naval)
D. Dennison, Army Personnel Research Establishment, Personnel Psychological Division, England (2)
Science 3 (RAF), Lacon House, England
Admiralty Research Establishment, Applied Psychology Unit, Teddington, England (Richard Gregory)
1 Psychological Research Unit, NBH 3-44, Australia
Directorate of Psychology - AF, Department of Defense (Air for CE), Australia
Navy Psychology, Australia (2)
Defense Psychology Unit, Defense HQ, New Zealand (2)
Defense Technical Information Center (DDAC) (2)

SUPERINTENDENT
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5100

0142